PENGEMBANGAN ALGORITMA PENJADUALAN PRODUKSI *JOB SHOP* UNTUK MEMINIMUMKAN TOTAL BIAYA *EARLINESS* DAN *TARDINESS*

Dian Retno Sari Dewi

Jurusan Teknik Industri, Universitas Katolik Widya Mandala Surabaya
Jl. Dinoyo no. 22 – 24, Surabaya
email: dessi@mail.wima.ac.id

ABSTRACT

This paper develops job shop production scheduling using Non Delay algorithm through forward and backward-forward algorithm to minimize total earliness and tardiness costs. Backward approach has some disadvantages, such as, if the job is scheduled in backward, there is a possibility that the infeasible situation occurs, in which the job is scheduled at $t < 0$. This paper used hypothetic data generated randomly. This job shop scheduling algorithm development was validated using LINDO software to check the effectiveness heuristic method, compared with the optimization method. The validation proves that the result of backward-forward scheduling method is better than the result of forward scheduling method.

Keywords: job shop scheduling, forward non delay, backward-forward non delay

Pendahuluan

Penjadualan didefinisikan sebagai pengalokasian sumber-sumber daya selama suatu rentang waktu untuk melakukan sekumpulan tugas (Baker, 1990). Tujuan penjadualan adalah mengoptimasi penggunaan sumber daya sehingga tujuan produksi dapat tercapai (Narasimhan, dkk, 1985). Dalam penjadualan berbasis waktu dikenal keterlambatan positif (*tardiness*) dan keterlambatan negatif (*earliness*), bila dikaitkan dengan aturan penjadualan berbasis biaya keterlambatan positif menimbulkan biaya *tardiness* dan keterlambatan negatif menimbulkan biaya *earliness*. Biaya *tardiness* adalah biaya keterlambatan yang disebabkan suatu pekerjaan diselesaikan lebih dari batas waktu (*due date*) yang ditetapkan oleh konsumen, sedangkan biaya *earliness* adalah biaya yang disebabkan suatu pekerjaan diselesaikan lebih cepat dari batas waktu (*due date*) yang telah ditentukan oleh konsumen sehingga menimbulkan biaya inventori.

Tjandera (1992), melakukan penelitian penjadualan produksi menggunakan metode *forward non delay* dan *backward* untuk menyelesaikan masalah di lingkungan *job shop*. Ibnu Utama (1994), melakukan penelitian penjadualan *job shop* untuk meminimasi *earliness* dengan mempertimbangkan perawatan mesin. Sun dan Lin (1994), memberikan kerangka berpikir penyelesaian masalah *job shop* dengan pendekatan *backward*. Karena sebelumnya belum ada penelitian tentang penjadualan *job shop* yang meminimasi total biaya *earliness* dan *tardiness* secara bersama-sama, maka dicoba untuk mengembangkan algoritma penjadualan produksi *job shop* untuk meminimumkan total biaya *earliness* dan *tardiness* dengan metode *Non Delay* melalui pendekatan *forward* dan *backward*. Pendekatan *backward* mempunyai kekurangan, jika *job* dijadwalkan mundur akan memungkinkan terjadinya *infeasible*, yaitu suatu

keadaan dimana *job* dijadwalkan pada $t < 0$. Maka *job* yang *infeasible* akan dimajukan pada $t = 0$ dengan menggunakan algoritma *affected operation rescheduling*.

Tujuan

Tujuan penelitian ini adalah untuk:

1. Mengembangkan algoritma penjadualan *job shop* untuk meminimumkan total biaya *earliness* dan *tardiness* secara bersama-sama dengan metode *Non Delay* melalui pendekatan *forward* dan *backward-forward*.
2. Membandingkan antara metode *forward non delay* dengan metode *backward-forward non delay*.
3. Membandingkan aturan *priority rules* yang dipakai dalam metode *forward non delay* dan *backward-forward*.

Batasan Masalah

Batasan kajian yang digunakan dalam penelitian ini meliputi:

1. Program aplikasi penjadualan *job shop* ini hanya mampu untuk menjadwalkan maksimal 50 *job* dan 50 operasi.
2. *Priority rules* yang digunakan pada metode *forward non delay* adalah EDD (*Earliest Due Date*), SPT (*Shortest Processing Time*), dan S/OPN (*Slack per Operation*).
3. *Priority rules* yang digunakan pada metode *backward-forward non delay* adalah LDD (*Last Due Date*), LPT (*Longest Processing Time*), dan S/OPN (*Slack per Operation*).

Asumsi

1. Waktu transportasi diabaikan.
2. Waktu set-up mesin diabaikan.
3. Kedatangan *job* pada saat $t = 0$ untuk metode *forward non delay*.
4. Keadaan proses produksi diasumsikan dalam kondisi statis.
5. Tidak ada kegiatan lain yang dapat menyela berjalannya proses produksi.

Pengembangan Model

Metode forward non delay

Metode *forward non delay* atau penjadualan maju adalah penjadualan yang dilakukan searah dengan perubahan waktu, dimulai dari saat tersedianya *job* sampai akhir waktu penyelesaian *job*. Algoritma *forward non delay* adalah sebagai berikut :

1. Tahap inisiasi : identifikasi jumlah dan *routing job*.
2. Misalkan $t = 1$ dan P_t merupakan himpunan kosong dan S_t himpunan *task* tanpa pendahulu.
3. Cari R^* dimana $R^* = \min C_j \in S_t \{R_j\}$ dan m^* dimana R^* dilakukan.
4. Untuk setiap *task* $ij \in S_t$ yang memerlukan m^* dan $R_{ij} = R^*$ pilih *task* dengan *priority decision rules* yang diinginkan.
5. Lanjutkan ke tahap berikutnya, yaitu :
 - a. Tambahkan *task* yang terpilih pada P_t sehingga membentuk P_{st+1} .
 - b. Hilangkan *task* ij terpilih dari S_t dan bentuk S_{t+1} dengan cara menambahkan *successor*-nya apabila seluruh *task* pendahulunya sudah terjadwalkan (pengecualian untuk *task* terakhir).
 - c. Tambahkan t dengan 1.

- d. Perbaiki R_{ij} untuk seluruh *task* di S_{t+1} yaitu $R_{ij} = \max(\max x_{ij,ij}) \text{ Pred}(ij)$, $\max x_{ij}$ dimana *task* x_{ij} membutuhkan m , *task* ij P_{st} .
6. Jika masih ada *task* yang belum terjadualkan, lakukan langkah 3, sebaliknya stop.

Metode *backward-forward non delay*

Menurut Sun dan Lin, performansi penting dalam lingkungan *job shop* adalah performansi *due date* dan biaya persediaan. Namun disamping banyaknya kelebihan yang dimiliki oleh pendekatan *backward*, pendekatan *backward* memiliki kelemahan, yaitu jadual yang dihasilkan *infeasible*. Jadual *infeasible* ini dikarenakan output penjadualan *backward* yaitu *job release time* berada diluar rentang horizon perencanaan penjadualan [5]. Metode *backward-forward non delay* atau penjadualan mundur adalah penjadualan yang dilakukan berbalikan arah dengan perubahan waktu. Kondisi awal pada penjadualan mundur adalah *job due date*, dengan hasil akhir *job release time*.

Algoritma *backward-forward non delay* adalah sebagai berikut :

1. Tahap inisiasi : identifikasi jumlah dan *routing job*.
2. Menentukan nilai awal mesin sesuai dengan *due date job* yang dikerjakan di mesin tertentu.
3. Menentukan S_t masing-masing *job* dimulai dari operasi yang paling akhir.
4. Cari R^* dimana $R^* = \max C_j \in S_t \{R_j\}$ dan m^* dimana R^* dilakukan.
5. Pilih dan tentukan operasi $j \in S_t$ dimana operasi j dilakukan di m^*
 - a. Apakah operasi $j > 1$?
Pada saat pemilihan apakah operasi j yang berada dalam S_t dan dilakukan di m^* dimana R^* dilakukan lebih dari satu ? Jika tidak, lanjutkan ke langkah 6.
 - b. Jika ya maka, lakukan pemilihan operasi j dengan menggunakan *priority decision rules* yang diinginkan kemudian lanjutkan ke langkah 6.
6. Masukkan operasi j yang sudah terpilih pada P_{st+1} .
7. Hilangkan operasi j yang telah dijadualkan pada S_t .
8. Perbaharui *available time* untuk setiap mesin.
9. Jika masih ada *task* yang belum terjadualkan, lakukan langkah 4 lalu stop.
10. Jika hasil penjadualan *infeasible*, majukan semua *job* yang *infeasible* pada titik $t=0$. Geser semua *job* yang terpengaruh.
11. Set $i=1$, $g=1$, waktu selesai = 0, $devSt = 0$, $O = \Phi$, $aff = \Phi$, *job start* = *mcStart* = 0 untuk semua operasi.
12. $O[g]$ sebagai operasi sekarang, set *mcStart* = C_r ; g ditambah 1
NewStart sekarang = maks (*jobStart*, *mcStart*)
NewEnd sekarang = *newStart* sekarang + t_{so} .
13. Jika *job* sekarang tidak cocok dengan kumpulan operasi ter-*affected* pada kumpulan 'aff', ke langkah 14, jika tidak maka *reset aff* (v) sebagai berikut, kemudian ke langkah 15.
14. *NewStart aff* (v) = maks {*newStart* sekarang, *newStart aff* (v)}.
15. *NewEnd aff* (v) = *newStart aff* (v) + waktu proses *aff* (v).
16. Set $aff[i]$ = operasi sekarang; i ditambah 1.
17. Cari noj , jika noj ada dan *startTime* < *newEnd* sekarang, maka :
18. $O[g] = noj$ dan *jobStart* dari $O[g] = newEnd$ sekarang; $g+1$.

-
19. Cari nom, jika nom ada dan $startTime < newEnd$ sekarang, maka:
 20. $O[g] = nom$ dan $mcStart$ dari $O[g] = newEnd$ sekarang; $g+1$.
 21. Kurangkan operasi sekarang dari set O , dan tambahkan anggota baru O dari langkah 16 dan 17.
 22. Jika $O = \emptyset$ maka STOP, jika tidak cari noj dan nom baru dari set O yang ada sekarang.

Keterangan simbol dan istilah yang digunakan:

- Noj = operasi berikutnya dari *job*
 Nom = operasi berikutnya dari mesin
 JobStart = waktu mulai operasi yang dibatasi operasi sebelumnya dari *job*
 McStart = waktu mulai operasi yang dibatasi operasi sebelumnya dari mesin
 StartTime = waktu mulai operasi dari penjadualan yang lama
 EndTime = waktu selesai operasi dari penjadualan yang lama
 NewStart = waktu mulai operasi dari penjadualan yang baru
 NewEnd = waktu selesai operasi dari penjadualan yang baru
 DevSt = deviasi waktu mulai antara penjadualan yang lama dan penjadualan yang baru
 Aff = set operasi yang terpengaruh setelah rescheduling (dengan waktu mulai dan selesai yang baru)
 O = set operasi yang terpengaruh
 I = index dari aff
 G = index dari O

Pembahasan

Data yang digunakan adalah data hipotetik yang berisi urutan proses, mesin yang digunakan dan *due date*.

Metode *forward non delay*

Pengolahan data menggunakan program aplikasi *Microsoft Visual Basic*. *Priority rules* yang digunakan pada metode *forward non delay* adalah EDD (*Earliest Due Date*), SPT (*Shortest Processing Time*), dan S/OPN (*Slack per Operation*). Total biaya *earliness* dan *tardiness* dapat dilihat pada tabel 1.

Metode *backward-forward non delay-forward non delay*

Priority rules yang digunakan pada metode *backward-forward non delay* adalah LDD (*Last Due Date*), LPT (*Longest Processing Time*), dan S/OPN (*Slack per Operation*). Pendekatan *backward* mempunyai kekurangan, jika *job* dijadwalkan mundur akan memungkinkan terjadinya *infeasible*, yaitu suatu keadaan dimana *job* dijadwalkan pada $t < 0$. Maka dilakukan pendekatan *backward-forward* untuk memajukan *job* yang *infeasible* pada $t = 0$ dengan menggunakan algoritma *affected operation rescheduling*. Total biaya *earliness* dan *tardiness* dapat dilihat pada tabel 2.

Validasi dilakukan dengan membandingkan output keluaran algoritma pengembangan dengan metode optimasi. Hasil validasi dapat dilihat pada tabel 3.

Validasi dilakukan untuk mengetahui seberapa efektif metode heuristik dibandingkan metode optimasi. Formula yang digunakan dalam perbandingannya

adalah menggunakan *range*. Range adalah perbedaan output metoda optimasi dikurangkan dengan output metoda pengembangan dengan metoda forward non delay.

Rata-rata *range* metoda optimasi dengan metoda *forward non delay* :

1. Rata-rata *range* data dengan *due date* ketat (data ke 1 sampai dengan data ke 15) adalah 18.13.
2. Data dengan *due date* longgar (data ke 16 sampai dengan data ke 30) adalah 36.07.
3. Secara umum rata-rata *range* untuk keseluruhan data, baik data yang mempunyai *due date* ketat maupun longgar (data ke 1 sampai dengan data ke 30) adalah 27.01.

Tabel 1. Output metode *forward non delay*

Data	EDD	SPT	S / OPN	Data	EDD	SPT	S / OPN
1	24	24	13	16	20	20	20
2	76	51	36	17	14	14	14
3	27	19	26	18	20	15	15
4	36	36	36	19	15	15	15
5	79	58	54	20	52	53	47
6	91	64	41	21	80	80	67
7	119	118	77	22	94	88	70
8	13	28	24	23	72	59	51
9	51	58	23	24	62	33	48
10	17	17	17	25	74	82	67
11	79	33	22	26	93	85	88
12	6	8	6	27	22	22	22
13	15	15	15	28	47	47	47
14	24	11	11	29	57	63	57
15	7	7	7	30	51	8	8

Tabel 2. Output metode *backward-forward non delay-forward non delay*

Data	LDD	LPT	S / OPN	Data	LDD	LPT	S / OPN
1	12	12	12	16	0	0	0
2	57	240	8	17	2	2	40
3	9	65	38	18	20	18	2
4	27	27	32	19	12	12	12
5	161	58	30	20	19	57	9
6	49	123	40	21	31	66	58
7	98	98	72	22	30	66	50
8	57	47	50	23	11	11	28
9	20	68	25	24	160	83	26
10	0	70	29	25	20	52	35
11	2	4	12	26	56	56	28
12	5	5	5	27	0	0	0
13	12	12	12	28	19	42	39
14	8	21	21	29	20	20	13
15	6	6	6	30	0	0	0

Tabel 3. Validasi dengan metoda optimasi

Data	Metode optimasi	Metode <i>forward</i>	Metode <i>backward-forward non delay</i>	Data	Metode optimasi	Metode <i>forward</i>	Metode <i>backward-forward non delay</i>
1	0	13	12	16	0	20	0
2	8	36	8	17	0	14	2
3	9	19	9	18	2	15	2
4	2	36	27	19	0	15	12
5	24	54	30	20	9	47	9
6	20	41	40	21	15	67	31
7	25	77	72	22	5	70	30
8	1	13	47	23	1	51	11
9	18	23	20	24	3	33	26
10	0	17	0	25	4	67	20
11	2	22	2	26	8	85	28
12	5	6	5	27	0	22	0
13	0	15	12	28	19	47	19
14	0	11	8	29	11	57	13
15	4	7	6	30	0	8	0

Rata-rata *range* metoda optimasi dengan metode *backward-forward non delay* adalah :

1. Rata-rata *range* data dengan *due date* ketat (data ke 1 sampai dengan data ke 15) adalah 12.
2. Data dengan *due date* longgar (data ke 16 sampai dengan data ke 30) adalah 8.4.
3. Secara umum rata-rata *range* untuk keseluruhan data, baik data yang mempunyai *due date* ketat maupun longgar (data ke 1 sampai dengan data ke 30) adalah 10.2.

Dari hasil validasi untuk metode *forward non delay* dan metode *backward-forward non delay* di atas diketahui bahwa hasil penjadwalan metode *backward-forward non delay* lebih mendekati optimal dibandingkan dengan hasil penjadwalan dengan metode *forward non delay*. Hal ini dikarenakan penjadwalan dengan metode *backward-forward non delay* dimulai dari *due date* kemudian dijadualkan mundur mendekati $t=0$, sehingga akan meminimumkan total biaya *earliness* dan *tardiness*, karena penjadwalan dimulai dari *due date*.

Analisa Priority Rules

Metode *forward non delay* menggunakan tiga *priority rules* yaitu, EDD (*Earliest Due Date*), SPT (*Shortest Processing Time*), dan S/OPN (*Slack per Operation*). Secara umum, *priority rules* yang menghasilkan biaya terkecil adalah S/OPN. Perhitungan pada S/OPN menggunakan *due date* dan sisa operasi, sehingga *job* akan dijadualkan secara merata. Hal ini menyebabkan total biaya *earliness* dan *tardiness* menjadi lebih kecil. Sedangkan pada EDD dan SPT perhitungan tidak menggunakan menggunakan

due date dan sisa operasi, sehingga *job* tidak dijadualkan secara merata. Hal ini menyebabkan total biaya *earliness* dan *tardiness* menjadi lebih besar.

Metode backward-forward non delay menggunakan tiga *priority rules* yaitu, LDD (*Last Due Date*), LPT (*Longest Processing Time*), dan S/OPN (*Slack per Operation*). Secara umum, *priority rules* yang menghasilkan biaya terkecil adalah LDD dan S/OPN. Perhitungan pada LDD menggunakan *due date*, sehingga *job* dijadualkan mulai dari *due date* (tidak terlalu awal dan tidak terlalu lambat) ke depan. Hal ini menyebabkan total biaya *earliness* dan *tardiness* menjadi kecil. Perhitungan pada S/OPN menggunakan *due date* dan sisa operasi, sehingga *job* akan dijadualkan secara merata. Hal ini menyebabkan total biaya *earliness* dan *tardiness* menjadi lebih kecil. Sedangkan pada LPT perhitungan tidak menggunakan menggunakan *due date* dan sisa operasi, sehingga *job* tidak dijadualkan secara merata. Hal ini menyebabkan total biaya *earliness* dan *tardiness* menjadi lebih besar.

Kesimpulan

1. Hasil penjadualan metode *backward-forward non delay* lebih mendekati optimal dibandingkan dengan hasil penjadualan dengan metode *forward non delay* dalam hal meminimasi total biaya *earliness* dan *tardiness*.
2. *Priority rules* S/OPN memberikan hasil terbaik pada penerapan metode *forward non delay* dalam hal meminimasi total biaya *earliness* dan *tardiness*.

Priority rules LDD dan S/OPN memberikan hasil terbaik pada penerapan metode backward- forward non delay dalam hal meminimasi total biaya *earliness* dan *tardiness*.

Referensi

- Ansari, N. and Hou, E. *Computational Intelligence for Optimization*. Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, New Jersey 07102.
- Baker, K.R. 1990. *Introduction to Sequencing and Scheduling*. Partmouth College.
- Fogarty, D.W, Blackstone, J.H and Hoffman, T.R. 1991. *Production and Inventory Management*. Cincinnati, Ohio, South-Western Publishing Co.
- Narasimhan, S.L.; McLeavey, D.W.; and Billington, P.J. 1985. *Production planning and Inventory Control*, 2nd ed. Prentice Hall International.
- Sun D.; and Lin L 1994. A Dynamic Job Shop Scheduling Framework : A Backward Approach. *International Journal of Production Research*. Vol 32, No.4, 967-985.
- Tjandera. 1992. *Penjadualan Produksi Metode forward non delay dan Backward untuk Lingkungan Job Shop*. Tugas Akhir.
- Utama, I. 1994. *Penjadualan Job Shop untuk Meminimasi Earliness Dengan Mempertimbangkan Perawatan Mesin*". Tugas Akhir. Institut Teknologi Bandung.

Lampiran

Contoh data hipotetik.

1. Data 1

Waktu proses :

Job	Operasi			
	1	2	3	4
1	8	7	8	6
2	5	7	8	7
3	6	5	6	6

Mesin :

Job	Operasi			
	1	2	3	4
1	3	1	4	2
2	2	1	4	3
3	1	3	2	4

Due date :

Job	DD
1	38
2	36
3	37

Iterasi program metode *forward* untuk data 1 :

Stage 1

Waktu Mesin

(1)0 (2)0 (3)0 (4)0

(St) 1 1 3, (Cj) 0, (Tij) 8

(St) 2 1 2, (Cj) 0, (Tij) 5

(St) 3 1 1, (Cj) 0, (Tij) 6

Stage 2

Waktu Mesin

(1)6 (2)5 (3)8 (4)0

(St) 1 2 1, (Cj) 8, (Tij) 7

(St) 2 2 1, (Cj) 6, (Tij) 7

(St) 3 2 3, (Cj) 8, (Tij) 5

Stage 3

Waktu Mesin

(1)13 (2)5 (3)8 (4)0

(St) 1 2 1, (Cj) 13, (Tij) 7
(St) 2 3 4, (Cj) 13, (Tij) 8
(St) 3 2 3, (Cj) 8, (Tij) 5

Stage 4

Waktu Mesin

(1)13 (2)5 (3)13 (4)0

(St) 1 2 1, (Cj) 13, (Tij) 7
(St) 2 3 4, (Cj) 13, (Tij) 8
(St) 3 3 2, (Cj) 13, (Tij) 6

Stage 5

Waktu Mesin

(1)20 (2)19 (3)13 (4)21

(St) 1 3 4, (Cj) 21, (Tij) 8
(St) 2 4 3, (Cj) 21, (Tij) 7
(St) 3 4 4, (Cj) 21, (Tij) 6

Stage 6

Waktu Mesin

(1)20 (2)19 (3)28 (4)27

(St) 1 3 4, (Cj) 27, (Tij) 8

Stage 7

Waktu Mesin

(1)20 (2)19 (3)28 (4)35

(St) 1 4 2, (Cj) 35, (Tij) 6

Stage 8

Waktu Mesin

(1)20 (2)41 (3)28 (4)35

Biaya Earlines = 18

Biaya Tardines = 6

Total Biaya = 24